

IN THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application.

1. (currently amended) A method of managing a network switch having a processor card including a volatile memory and a processing unit in the processor card comprising:

detecting an error;

determining if the error is ignorable;

determining whether a threshold has been reached if the error is determined to be ignorable, the threshold corresponding to a number of hitless rebuilds that have occurred within an amount of time; and,

performing a hitless rebuild in the processor card if the threshold has not been reached, the performing a hitless rebuild comprising:

performing an initialization of the volatile memory; and,

protecting a portion of the volatile memory from access by the

processing unit during the initialization.

2. (canceled)

3. (previously presented) The method of claim 1, wherein the performing of the includes protecting a portion of the memory that contains a set of routing tables.

4. (previously presented) The method of claim 1, wherein the performing of the hitless rebuild further comprises protecting a portion of the memory that contains a set of state tables.

5. (previously presented) The method of claim 1, wherein the memory is accessed through a set of memory addresses and the performing of the hitless rebuild further comprises preventing the processing unit from accessing a predetermined set of memory addresses in the set of memory addresses.

6. (canceled)

7. (previously presented) The method of claim 1, further comprising setting the processing unit to enter into a degraded mode if the error is not ignorable and if the threshold has been reached.

8. (canceled)

9. (canceled)

10. (currently amended) An article comprising a machine readable medium having instructions stored thereon which, when executed, cause a method to be performed, the method comprising:

detecting an error;

determining if the error is an ignorable error;

determining whether a threshold has been reached if the error is determined to be an ignorable, the threshold corresponding to a number of hitless rebuilds that have occurred within an amount of time; and,

performing a hitless rebuild in a processor card if the threshold has not been reached, the performing a hitless rebuild comprising:

performing an initialization of a volatile memory;

protecting a portion of the volatile memory from access by the
processing unit during the initialization.

11. (canceled).

12. (previously presented) The article of claim 10, wherein the method further comprises:

protecting a portion of the memory that contains a set of routing tables.

13. (previously presented) The article of claim 10, wherein the method further comprises:

protecting a portion of the memory that contains a set of state tables.

14. (previously presented) The article of claim 10, wherein the memory further comprises:

preventing a processing unit from accessing a predetermined set of memory addresses in the set of memory addresses.

15. (canceled)

16. (previously presented) The article of claim 10, wherein the method further comprises:

setting of the processing unit to enter into a degraded mode if the threshold has been reached.

17. (canceled)

18. (currently amended) A method, comprising:

within a networking hardware apparatus:

operating a card having volatile memory space partitioned at least into a protected memory region and a non protected memory region, said operating not:

~~discouraging the occurrence~~ diminishing the likelihood of an error within said protected memory region any more than the ~~occurrence~~ likelihood of an error within said non protected memory region is ~~discouraged~~ diminished;

experiencing an error; and,

re-initializing software that is executed on said card, said re-initializing in response to said error, said card comprising a processor that executes said software, said protected memory region storing information that can be used by said

processor to execute said software after said software has been re-initialized, said information ~~further~~ comprising routing table information, said re-initializing not deleting said routing table information from said protected memory region, said re-initializing deleting other information from said non protected memory region.

19. (canceled)

20. (canceled)

21. (currently amended) The method of claim ~~20~~ 18 wherein said protected memory region further comprises a segment of random access memory, said not deleting further comprising said processor not accessing said segment during said re-initializing for purposes of clearing said segment during said re-initializing.

22. (currently amended) The method of claim ~~20~~ 18 wherein said protected memory region further comprises a segment of random access memory and said card comprises a memory management unit to manage said protected memory region, said not deleting further comprising said memory management unit not allowing access to said segment so as to prevent said segment from being cleared during said re-initializing.

23. (currently amended) A method, comprising:
within a networking hardware apparatus:

operating a card having volatile memory space partitioned at least into a protected memory region and a non protected memory region, said operating not:

~~discouraging the occurrence~~ diminishing the likelihood of an error within said protected memory region any more than the ~~occurrence~~ likelihood of an error within said non protected memory region is ~~is encouraged~~ diminished;

experiencing an error; and,

re-initializing software that is executed on said card, said re-initializing in response to said error, said card comprising a processor that executes said software, said protected memory region storing information that can be used by said processor to execute said software after said software has been re-initialized, said information ~~further~~ comprising state table information, said state table information further comprising the status of an interface card located within said networking hardware apparatus, said re-initializing not deleting said status from said protected memory region, said re-initializing deleting other information from said non protected memory region.

24. (canceled)

25. (canceled)

26. (currently amended) The method of claim ~~25~~ 23 wherein said protected memory region further comprises a segment of random access memory, said not

deleting further comprising said processor not accessing said segment during said re-initializing for purposes of clearing said segment during said re-initializing.

27. (currently amended) The method of claim 25 23 wherein said protected memory region further comprises a segment of random access memory and said card comprises a memory management unit to manage said protected memory region, said not deleting further comprising said memory management unit not allowing access to said segment so as to prevent said segment from being cleared during said re-initializing.

28. (currently amended) An apparatus, comprising:
a card for use in a networking hardware apparatus, said card comprising a processor to execute re-initializable software, said card comprising volatile memory, said volatile memory comprising a protected memory region and a non protected memory region, said protected memory region to store information that can be used by said processor to execute said re-initializable software after said software has been re-initialized, said information further comprising routing table information, said re-initializable software comprising instructions for re-initializing said software in response to an error, said instructions executable by said processor, said routing table information not being deleted from said protected memory region during said re-initializing, said card not comprising:

error preventive packaging that surrounds said protected memory region but not said non protected memory region.

29. (canceled)

30. (canceled)

31. (currently amended) The apparatus of claim ~~30~~ 28 wherein said protected memory region further comprises a segment of random access memory, said processor not accessible to said segment during said re-initializing so as to prevent said protected memory region from being cleared during said re-initializing.

32. (currently amended) The apparatus of claim ~~30~~ 28 wherein said protected memory region further comprises a segment of random access memory and said card further comprises a memory management unit, said memory management unit to not allow said segment to be accessed during said re-initializing for purposes of clearing said protected memory region.

33. (currently amended) An apparatus, comprising:

a card for use in a networking hardware apparatus, said card comprising a processor to execute re-initializable software, said card comprising volatile memory, said volatile memory comprising a protected memory region and a non protected memory region, said protected memory region to store information that can be used by said processor to execute said re-initializable software after said software has been re-initialized, said information ~~further~~ comprising state table information, said

state table information further comprising the status of an interface card located within said networking hardware apparatus, said re-initializable software comprising instructions for re-initializing said software in response to an error, said instructions executable by said processor, said status not being deleted from said protected memory region during said re-initializing, said card not comprising:

error preventive packaging that surrounds said protected memory region but not said non protected memory region.

34. (canceled)

35. (canceled)

36. (currently amended) The apparatus of claim ~~35~~ 33 wherein said protected memory region further comprises a segment of random access memory, said processor not accessible to said segment during said re-initializing so as to prevent said protected memory region from being cleared during said re-initializing.

37. (previously presented) The apparatus of claim ~~35~~ 33 wherein said protected memory region further comprises a segment of random access memory and said card further comprises a memory management unit, said memory management unit to not allow said segment to be accessed during said re-initializing for purposes of clearing said protected memory region.

38. (currently amended) An article comprising a machine readable medium having instructions stored thereon which, when executed, cause a method to be performed, the method comprising:

re-initializing software that is executed on a card within a networking hardware apparatus, said re-initializing in response to an error, said card comprising a processor that executes said software, said card comprising a volatile memory that stores information that can be used by said processor to execute said software after said software has been re-initialized, said information ~~further~~ comprising routing table information, said re-initializing not deleting said routing table information from said memory because said instructions are written to be unable to invoke memory addresses for a segment of said memory where said routing table information resides so as to make said segment of said memory inaccessible during said re-initializing.

39. – 42. (canceled)

43. (currently amended) The article of claim 42 38 wherein said ~~random access~~ volatile memory further comprises dynamic random access memory.

44. (currently amended) An article comprising a machine readable medium having instructions stored thereon which, when executed, cause a method to be performed, the method comprising:

re-initializing software that is executed on a networking device card, said re-initializing in response to an error, said networking device card comprising a processor that executes said software, said networking device card comprising volatile memory that stores information that can be used by said processor to execute said software, said information ~~further~~ comprising state table information, said re-initializing not deleting said state table information from said memory because said instructions are written to be unable to invoke memory addresses for a segment of said memory where said routing table information resides so as to make said segment of said memory inaccessible during said re-initializing.

45. – 48. (canceled)

49. (currently amended) The article of claim 48 44 wherein said ~~random access~~ volatile memory further comprises dynamic random access memory.

50. (currently amended) An apparatus, comprising:

means for re-initializing software that is executed on a card located within a networking hardware apparatus, said re-initializing in response to an error, said card comprising a processor to execute said software, said networking device card comprising volatile memory, said volatile memory comprising a non protected memory region and a protected memory region, said protected memory region to store information that can be used by said processor to execute said software after said software has been re-initialized, said information ~~further~~ comprising routing

table information, said means for re-initializing further comprising means for not deleting said routing table information from said protected memory region during said re-initializing, said card not comprising:

error preventive packaging that surrounds said protected memory region but not said non protected memory region.

51. – 53. (canceled)

54. (currently amended) The apparatus of claim ~~53~~ 50 wherein said [random access] memory further comprises dynamic random access memory.

55. – 59. (canceled)

60. (currently amended) A method, comprising:

within a networking hardware apparatus:

operating a card having volatile memory space partitioned at least into a protected memory region and a non protected memory region, said operating not:

~~discouraging the occurrence~~ diminishing the likelihood of an error within said protected memory region any more than the ~~occurrence~~ likelihood of an error within said non protected memory region is ~~discouraged~~ diminished;

experiencing an error; and,

re-initializing software that is executed on a card located within said networking hardware apparatus, said re-initializing in response to said error, said card comprising a processor that executes said software, said protected memory region storing information that can be used by said processor to execute said software after said software has been re-initialized, said information ~~further~~ comprising state table information, said state table information further comprising network topology information, said re-initializing not deleting said network topology information from said protected memory region, said re-initializing deleting other information from said non protected memory region.

61. (canceled)

62. (canceled)

63. (currently amended) The method of claim 62 60 wherein said protected memory region further comprises a segment of random access memory, said not deleting further comprising said processor not accessing said segment during said re-initializing for purposes of clearing said segment during said re-initializing.

64. (currently amended) The method of claim 62 60 wherein said protected memory region further comprises a segment of random access memory and said card comprises a memory management unit to manage said protected memory region, said not deleting further comprising said memory management unit not

allowing access to said segment so as to prevent said segment from being cleared during said re-initializing.

65. (currently amended) An apparatus, comprising:

a card for use in a networking hardware apparatus, said card comprising a processor to execute re-initializable software, said card comprising volatile memory, said volatile memory comprising a protected memory region and a non protected memory region, said protected memory region to store information that can be used by said processor to execute said re-initializable software after said software has been re-initialized, said information further comprising state table information, said state table information further comprising network topology information, said re-initializable software comprising instructions for re-initializing said software in response to an error, said instructions executable by said processor, said network topology information not being deleted from said protected memory region during said re-initializing, said card not comprising:

error preventive packaging that surrounds said protected memory region but not said non protected memory region.

66. (canceled)

67. (canceled)

68. (currently amended) The apparatus of claim ~~67~~ 65 wherein said protected memory region further comprises a segment of random access memory, said processor not accessible to said segment during said re-initializing so as to prevent said protected memory region from being cleared during said re-initializing.

69. (currently amended) The apparatus of claim ~~67~~ 65 wherein said protected memory region further comprises a segment of random access memory and said card further comprises a memory management unit, said memory management unit to not allow said segment to be accessed during said re-initializing for purposes of clearing said protected memory region.